



Topology-based hierarchical scheduling using deficit round robin

Flow protection and isolation for triple play service

Yu, Hao; Yan, Ying; Berger, Michael Stubert

Published in:

First International Conference on Future Information Networks, 2009. ICFIN 2009

Link to article, DOI:

[10.1109/ICFIN.2009.5339606](https://doi.org/10.1109/ICFIN.2009.5339606)

Publication date:

2009

Document Version

Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

Citation (APA):

Yu, H., Yan, Y., & Berger, M. S. (2009). Topology-based hierarchical scheduling using deficit round robin: Flow protection and isolation for triple play service. In *First International Conference on Future Information Networks, 2009. ICFIN 2009* (pp. 269-274). IEEE. <https://doi.org/10.1109/ICFIN.2009.5339606>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Topology-based Hierarchical Scheduling Using Deficit Round Robin: Flow Protection and Isolation for Triple Play Service

Hao Yu, Ying Yan, and Michael S. Berger

Department of Photonics Engineering
Technical University of Denmark
Kgs. Lyngby, Denmark
{haoyu, yiya, msbe}@fotonik.dtu.dk

Abstract— This paper proposes a topology-based hierarchical scheduling scheme using Deficit Round Robin (DRR). The main idea of the topology-based hierarchical scheduling is to map the topology of the connected network into the logical structure of the scheduler, and combine several token schedulers according to the topology. The mapping process could be completed through the network management plane or by manual configuration. Based on the knowledge of the network, the scheduler can manage the traffic on behalf of other less advanced nodes, avoid potential traffic congestion, and provide flow protection and isolation. Comparisons between hierarchical scheduling, flow-based scheduling, and class-based scheduling schemes have been carried out under a symmetric tree topology. Results have shown that the hierarchical scheduling scheme provides better flow protection and isolation from attack of malicious traffic. This is significant for IPTV services in Carrier Ethernet networks.

Keywords—Carrier Ethernet; IPTV; topology-based hierarchical scheduling; traffic management

I. INTRODUCTION

The pressures from the shifting entertainment and communication needs of residential customers are pushing operators to upgrade their networks to have the capabilities of delivering voice, video, and data services, also known as triple play services. Most voice, video, and data services used to be provided by separated networks, such as Public Switched Telephone Network (PSTN), the cable television network, and the Internet. The tendency of today is to integrate the services on a single network. In such a network, video broadcasting/multicasting and Video on Demand (VoD) services on IP networks (also known as IPTV services) will significantly increase the traffic load. Without a well-designed traffic management scheme, the quality of IPTV services may be guaranteed at the cost of the user experience degradation of Voice-over-IP (VoIP) service and broadband Internet access. Thus, various Quality-of-Service (QoS) requirements of each traffic type must be guaranteed by the integrated network using a well-designed traffic management scheme. Carrier Ethernet is becoming an important candidate to the Metro network as a replacement for synchronous optical network/digital hierarchy

(SONET/SDH) [1]. Traffic management of Ethernet switches has thus become an unavoidable research topic.

A considerable amount of research has been carried out to investigate the packet scheduling algorithm, which is one significant part of QoS provisioning. In some real-time services such as IPTV, continuous packet flows of a certain type of traffic are transmitted from sources to destinations. Packet schedulers should be able to fairly handle traffic flows and have low time-complexity of packet selection and forwarding. Deficit Round Robin (DRR) scheduling algorithm is capable of providing low complexity and near-perfect flow isolation [2]. The previous study in [3] has shown that flow-based scheduling scheme compared with the class-based is able to treat traffic flows separately, and thus provides better per flow protection using DRR in Carrier Ethernet transport networks. For network operators to provide end-to-end flow protection, it is needed to upgrade the whole network with such flow-based scheduling nodes. This results in a considerable capital investment and a long deployment period. Keeping the preferable features of Carrier Ethernet such as cost-efficiency and low complexity, this paper proposes a topology-based hierarchical scheduling scheme. The term, hierarchical scheduling, has been mentioned and discussed actively in other researchers' work. In [4], [5], [6], hierarchical scheduling is mainly discussed as an improvement to the traditional DRR for a single network node.

The contribution of this paper is a topology-based hierarchical scheduling scheme for carrier class Ethernet with simple Ethernet switches. Given the detailed topology of the network, including the output bandwidth of each node for instance, the topology-based hierarchical scheduler can combine several logical token schedulers to form a map of the actual topology with specific token rate. Tokens are generated for each arriving packet, carrying important scheduling information such as packet length, source and destination IDs, and so forth, and then are stored in token queues. By executing DRR among token queues and controlling the token rate, the topology-based hierarchical scheduler can schedule packets on behalf other interior nodes, avoid traffic congestion, provide per flow protection, and thus guarantee QoS requirements. The use of tokens to schedule packets aims at increasing the

This work was supported in part by the Danish National Advanced Technology Foundation and is carried out within the research project HIPT (High quality IP network for IPTV and VoIP).

processing speed in multi-level hierarchical scheduler, since tokens are usually short in length compared to actual packets. The concept of token scheduling is also discussed and used in [7].

Besides, the topology-based hierarchical scheduling scheme could be beneficial when it comes to network deployment and investment. Some less advanced nodes could remain in the network, giving the responsibility of traffic management to the topology-based hierarchical scheduling node. For simplicity, we will use the term “hierarchical scheduling” in this paper.

The rest of this paper is arranged as follows. In Section II, we explain the advantages of DRR and the reason of choosing it as a basic scheduling scheme. In Section III, the concept of hierarchical scheduling is illustrated and the benefit is discussed. Section IV presents the evaluation based on simulation results. Finally we conclude in Section V.

II. SCHEDULING ALGORITHMS COMPARISON

Scheduling algorithms are used in a router/switch design in order to attain the QoS requirement and fairly allocate limited resources among traffic flows. A significant amount of research has contributed to the development of scheduling algorithms, and they can be mainly divided into two categories: timestamp-based scheduling (also known as sorted-priority scheduling) and frame-based scheduling.

Timestamp-based schedulers maintain a global virtual time to emulate the ideal Generalized Processor Sharing (GPS) [8]. Arriving packets are marked with timestamps which are generated through the virtual machine. The timestamps are used by the scheduler to determine the order of packet departure. This category includes Weighted Fair Queuing (WFQ) [9], Worst-case Fair Weighted Fair Queuing (WF²Q) [10], Self-Clocked Fair Queuing (SCFQ) [11], and Start-time Fair Queuing (SFQ) [12]. These timestamp-based schedulers can approximate the fluid model well and provide good fairness and low latency. A main drawback is the high complexity involved in computing.

Frame-based schedulers serve packets in a round robin way. During each round a flow receives at least one transmission opportunity [13]. This category includes Deficit Round Robin (DRR) [2] and Elastic Round Robin (ERR) [14]. These schedulers do not need to calculate the virtual time, and thus have low complexity and the design of such frame-based schedulers will be fairly simple. DRR is one of the early frame-based scheduling algorithms proposed to overcome the unfairness. It has been concluded in [2] that the DRR provides near-perfect isolation at low implementation cost and can be combined with other fair queuing algorithms to offer better latency bounds. Low complexity is a significant factor to a router/switch design, especially for a high speed link. Thus, DRR is selected as a basic scheduling algorithm for this work. Together with advanced buffer management, DRR can support sufficient QoS differentiation between flows and can guarantee that any maliciously behaving flow does not affect the QoS performance of other conforming traffic flows. The work in this paper is an extension of the previous work in [3]. An intelligent topology-based scheduler using flow-based DRR is introduced.

III. TOPOLOGY-BASED HIERARCHICAL SCHEDULING

Topology-based hierarchical scheduling is motivated by the need of a centralized intelligence scheme to overcome potential limitations of distributed intelligence in a network. It has been shown in [3] that flow-based scheduling scheme, compared to the class-based, has advantages in terms of flow isolation and protection. Thus, using the idea of flow-based schedulers to build the architecture of a hierarchical scheduler will retain the preferable features.

A. Motivation of Hierarchical Scheduling

Distributing intelligence in a network to provide QoS, which means that all the nodes are equipped with an advanced packet scheduling scheme, will place burdens on the operator. Such a process of upgrading a whole large network inevitably requires a long deploying time and a substantial financial investment. A management plane which configures and manages the nodes is also necessary. Distributed intelligence thus may not be cost-efficient enough in some cases.

Centralized intelligence can be considered as an alternative to offer QoS guarantee. Instead of assigning intelligence, in terms of an advanced scheduling scheme, buffer management, and so forth, to all the nodes in the network, the operator could choose to move it to the edge. One good example could be the ingress and egress router in a Multiprotocol Label Switching (MPLS) network. Typically, the MPLS label is attached to an IP packet at the ingress router and removed at the egress router, while label swapping is performed at the intermediate routers.

The concept of centralized intelligence requires that the hierarchical scheduler is aware of the network topology and potential congestion points in the network. In this paper, we assume that the learning process is completed before traffic dissemination. Based on the network information, the hierarchical scheduler will be able to schedule traffic on behalf of other simple nodes and avoid potential traffic congestion in the network. Take a tree topology network for instance, the nodes at the lower levels may only support lower link rate while the ones at the upper level could forward packet at a higher rate. The edge node, located at the highest level, rearranges the order of the incoming packets from the source and sends them down to ensure that no overflow occurs.

B. Outline of Flow-based Scheduling

The main concept of the flow-based scheduling scheme discussed in [3] is to classify the packets of the same traffic type into different flows one step further based on source-destination ID pairs. By such further classification, the flow-based scheduler can provide per flow scheduling and protection.

A demonstration of a flow-based scheduler structure for one output port is shown in Fig. 1. As the class-based scheduler, the flow-based scheduler uses DRR scheduling algorithm between each type of traffic. Within each queue for a type of traffic, subqueues are generated to distinguish packets of different sources. Fig. 1 shows an example where N sources sending three types of traffic to a destination and the number of flows in this case thus becomes $3N$. Within each queue, a subscheduler is used to select a packet based on DRR to the

central scheduler. Rearranging the order of packets using this dual-level DRR scheduling scheme guarantees that flows are protected against any misbehaving traffic, and delay or jitter spreading across flows are diminished. Furthermore, the dual-level DRR structure reduces complexity to a large extent.

C. Architecture of Hierarchical Scheduling

Based on the structure of the flow-based scheduler illustrated in Fig. 1, and the same concept to protect flows, the hierarchical scheduler combines multiple token schedulers to fulfill its QoS requirement. Topology-based hierarchical scheduler, as the name indicates, learns the actual network topology to form its inner logical structure. The learning process can be completed by the network management plane or manual configuration. IPTV distribution usually follows a tree topology for multicast [15]. Thus, in this paper we assume an example of tree topology network shown in Fig. 2 to demonstrate how the topology-based hierarchical scheduler forms its inner structure. However, the proposed hierarchical scheduling scheme is not limited to the specific network topology shown in Fig. 2. As long as the potential congestion points in the network are known, the scheduler will be able to manage the traffic so as to avoid congestion under other topologies, such as symmetric tree, asymmetric tree, star, ring and so forth.

A simple balanced binary tree topology network is shown in Fig. 2. The hierarchical scheduler node is located at the root of the tree and is labeled as HS. It has two branches separately connected with a simple node. The two simple nodes, labeled as D40 and D41, are placed in Level 4. Under Level 4, there are other three levels, forming a 4-level hierarchical network. Node output link rate is the same for each level, but from the upper one to the lower, the link rate is reduced by half. Nodes from Level 4 to Level 2 represent simple switches, lacking the functionality of flow protection, traffic management, and buffer management. The 16 nodes, shown as S00 to S15 on Level 1, represent the end users.

IPTV traffic flows to the 16 end users travel through node HS and further down to the remaining part of the network. As explained previously, the intermediate nodes from Level 4 to Level 2 do not have the ability to schedule traffic flows but simply reject packets if the temporary storage buffer overflows.

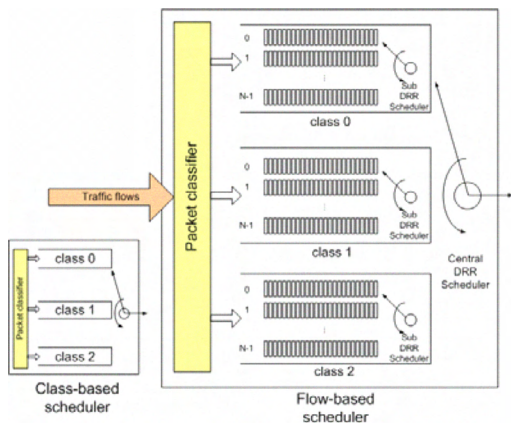


Figure 1. A flow-based scheduler is derived from a class-based scheduler by further dividing the queues into several subqueues based on flow ID.

It can be seen as a First-In-First-Out (FIFO) queue with limited buffer size for each output link. Thus it is node HS's accountability to take care of the traffic. Node HS performs DRR scheduling algorithm between each flow and ensure that malicious traffic will not affect other conforming one. Besides, the hierarchical scheduler should control the output throughput of every flow so that packets will not be discarded on the path to the end users.

Fig. 3 demonstrates the schematic structure of a hierarchical scheduler for one output port of Node HS connecting half of the tree in Fig. 2. Traffic flows enter the scheduler from the left. In order to demonstrate the scheme, two types of traffic and two source IDs are shown in the figure, i.e. class i and j , source 0 and 1 . Packets are sorted into queues by the packet classifier based on the destination-source ID pair and the type of traffic, and are stored in the packet queue memory. Tokens are generated simultaneously for each arriving packet and are forwarded to the token queues. A token should carry scheduling information for its corresponding packet, such as packet weight, destination-source ID, and type of traffic. The packet weight is a value in proportion to the actual packet length. It is used by the token scheduler as a virtual packet length to control the packet transmission rate. Packets bound to one destination will generate tokens which are further sorted into two queues by the token queue system based on the type of traffic.

The hierarchical scheduler forms four levels of token schedulers according to half of the tree in Fig. 2, denoted as S_1 , S_2 , S_3 , and S_4 . The numbers in the bracket indicate the destinations that the scheduler is in charge of. Each token scheduler uses DRR scheduling algorithm and has a token transmission rate reduced by half from Level 4 to Level 1, as denoted $8N$, $4N$, $2N$, and N in Fig. 3. The token rate ratio corresponds to the link rate of each level in Fig. 2. For an example, $S_4(0-7)$ schedules traffic for the link between HS to D40, and $S_3(0-3)$ schedules traffic for the link between D40 to D30. cS_x schedulers select tokens among different token queues of the same traffic type x . $TQ(s, i, d)$ stands for the Token Queue for packets of $\langle \text{source } s, \text{ type } i, \text{ destination } d \rangle$. Each token queue is allocated a deficit counter (DC). Owing to the hierarchical structure, each token scheduler is also allocated a DC, except the top-level scheduler $S_4(0-7)$.

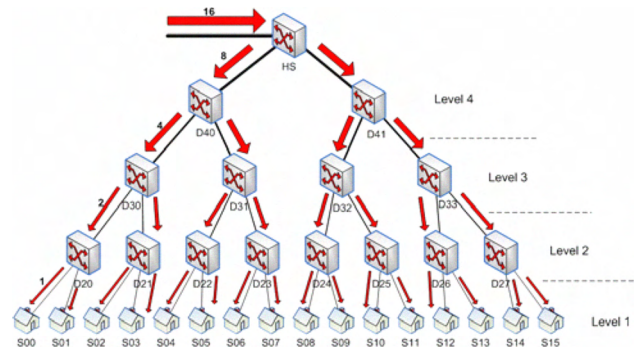


Figure 2. An example of a symmetric binary tree topology network, with link rate reduced by half from the upper level to the end. Hierarchical scheduling node is located at the edge of the network.

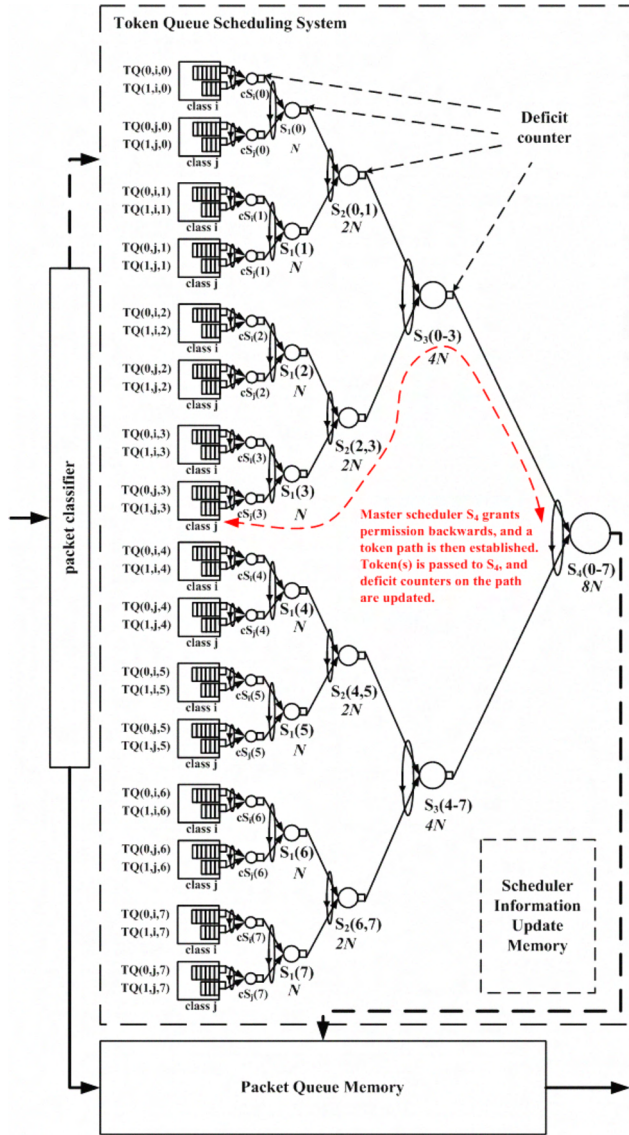


Figure 3. The schematic structure of topology-based hierarchical scheduler for one output port (half of the tree). Each token scheduler uses DRR algorithm and exchanges information in a shared memory.

The scheduler cS_x executes the DRR algorithm on the backlogged TQs of traffic type x . Preliminary decisions of token selection are made and the schedulers turn into a backlogged status. On the next level, the S_1 scheduler executes DRR algorithm on the backlogged cS_x schedulers and make its preliminary decision on which cS_x scheduler will be selected, and then turns into a backlogged status. The S_2 scheduler executes DRR algorithm among backlogged S_1 schedulers and turns backlogged after a decision is made. Then the S_3 scheduler does the same process, making a preliminary selection and turning backlogged. Finally the master scheduler S_4 chooses a backlogged S_3 scheduler based on DRR and grants permission. Permission is granted backwards from level to level and a token path will then be established in the end. Necessary information such as updating DCs, status, and feedback of schedulers are shared and exchanged in a memory. As the example shown in Fig. 3, a token path is established and the token(s) will be passed to the master scheduler. Based on

the information carried by the token(s), the top-level scheduler will send out the packet(s) from packet queue memory accordingly.

To avoid congestion occurs in the network, token schedulers control the packet transmission rate by the token rate and the packet weight. Packet Weight (PW) is a function of the actual packet length and is calculated by the packet classifier for each arriving packet. In the token scheduling system, a virtual packet transmission time is calculated as the packet weight divided by the token rate. The description of the notations and the PW function is as follows: l : the packet length stored in the Ethernet MAC header; f_w : the configurable packet weight factor. Since the token rate corresponds to the actual transmission rate of the node in the network, it is calculated as a product of the weight factor and the actual link rate. The description of the notations and token rate calculation is as follows: N : the basic token rate; R : the link rate of the end node. The network operator can modify the packet weight factor to control the granularity of token rates in order to adapt the switch to the network.

$$PW(l) = l \cdot f_w \quad (1)$$

$$N = f_w \cdot R \quad (2)$$

The time between two token selections is the sum of the packet weight of the token(s) divided by the token rate of the scheduler. By this mean, the packet transmission rate is controlled so that packets are transmitted within the capacity of the nodes in the network, and thus traffic congestion can be avoided.

It is important to mention that the structure of the hierarchical scheduler is not limited to the example of Fig.3 but can be reconfigured. If the topology is an asymmetric tree or a star for instance, the token schedulers will be reorganized and the logical structure will be configured accordingly.

The hierarchical scheduler is a linear combination of several DRR schedulers. Each DRR scheduler has a time complexity of $O(1)$ [1]. If the hierarchical scheduler has L hierarchies or levels, it needs to establish a token path in L steps and thus the time complexity will be $O(L)$. When $L = 1$, the scheduler will become a single DRR scheduler of which the time complexity is $O(1)$.

IV. PERFORMANCE EVALUATION

We evaluate the performance of the topology-based hierarchical scheduler by simulations in OPNET [16]. Networks with flow-based and class-based scheduling schemes are compared with the hierarchical scheduling in terms of flow isolation and protection.

A. Network Topology and Traffic Parameter Setup

Three networks are provided with the same 16 traffic flows. Each flow is bound to a destination. Each network has the same topology as shown in Fig. 2 and is built by one of the scheduling schemes. 16 Traffic flows are sent to these three networks simultaneously so that comparable results can be obtained. A normal traffic flow is configured to have an average bandwidth of 9 Mbps, which is similar to the

bandwidth needed by a high definition IPTV channel. 16 flows respectively bound to 16 destinations are sent to three networks simultaneously. The link speed is reduced by half for each level as explained in previous section. For the end user, the link supports up to 10 Mbps transmission rate.

To evaluate the flow protection and isolation ability of the networks, a highly bursty traffic flow is introduced for a certain period of time. The impact to the normal flow is then observed at the destination. The highly bursty traffic flow has a higher average bandwidth than a normal flow.

B. Simulation Results Analysis

The simulation lasts for 60 seconds and the highly bursty traffic flow is introduced from 10 to 20 second. In the networks shown in Fig. 4, the highly bursty flow is bound to S01. The flow to S00 is observed because it is most affected by the highly bursty flow.

In Fig. 5 the comparison between class-based, flow-based and hierarchical scheduling under the malicious flow attack is presented. The bandwidth of the highly bursty traffic is 9.5% more than the normal flow. Since the class-based scheduling scheme cannot distinguish different flows of the same traffic type, the normal flow is affected the most in terms of increase in end-to-end delay. Flow-based and hierarchical scheduling schemes are both capable of flow isolation, and thus the end-to-end delay of the normal flow increases slightly. Class-based scheduling scheme, under the malicious flow attack, performs the worst, and thus the comparison will be carried out between the flow-based and the hierarchical scheduling schemes.

In Fig. 6, the bandwidth of the highly bursty traffic flow is increased to be 67% more than a normal flow bandwidth. The affected end-to-end delay of the normal flow bound to destination S00 is presented. A comparison between the flow-based scheduling and the hierarchical scheduling is presented in this figure. The highest end-to-end delay of the flow-based scheduling network is increased up to around 4.5 ms, while the delay of the network using hierarchical scheduling scheme is increased to around 3.0 ms at most. After the highly bursty flow stops, both end-to-end delays are restored to the normal level. The hierarchical scheduling obviously has better performance than the flow-based one in terms of flow protection.

To further investigate the flow protection and isolation ability of the two scheduling schemes, i.e. flow-based scheduling and hierarchical scheduling, several experiments under different traffic load of the highly bursty flow are carried out. The average end-to-end delay of the affected period, during which the highly bursty flow is introduced, is measured for each circumstance. The comparison result is shown in Fig. 7.

The bandwidth of the highly bursty flow bound to destination S01 is increased from 10 to 16 Mbps. Both two schemes show similar average end-to-end delay under the bandwidth of 10 Mbps. This is because the switches in both networks still have enough capacity. Once the highly bursty flow increases the bandwidth more than the maximum limit, congestion will occur and consequently cause an addition to the

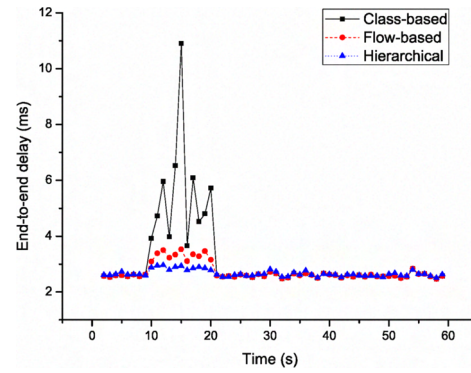


Figure 4. Comparison between class-based, flow-based and hierarchical scheduling in terms of traffic delay when a non-conforming flow appears. Bandwidth of the highly bursty traffic is 9.5% more than a normal flow.

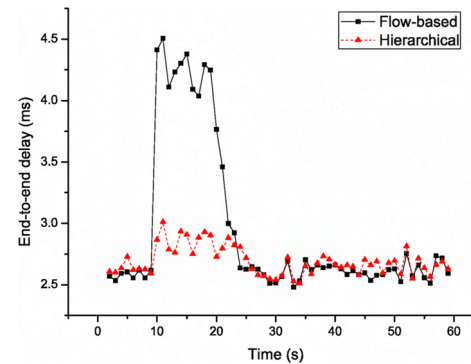


Figure 5. Comparison between flow-based and hierarchical scheduling in terms of traffic end-to-end delay when the load of a highly bursty flow increases. Bandwidth of the highly bursty traffic is 67% more than a normal flow.

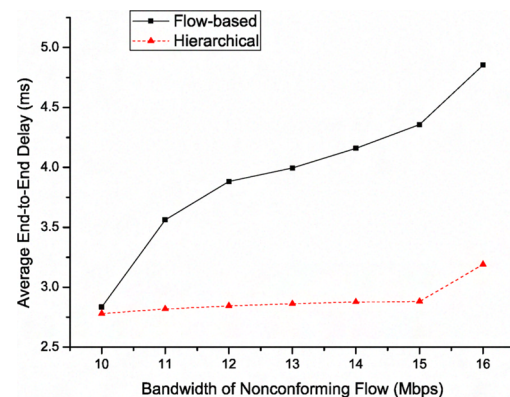


Figure 6. Comparison between flow-based and hierarchical scheduling in terms of average traffic delay of the affected period as the load of a highly bursty flow increases.

average end-to-end delay of the normal flow. The curve of the hierarchical scheduling scheme, compared to the flow-based,

remains stable, which indicates that the affected end-to-end delay of the normal flow is within a lower range of increase.

The improvement should be credited to centralizing network intelligence in the edge node. Potential congestion or any malicious attack is handled by the scheduler inside the node. Necessary internal resources are arranged and utilized by the node to diminish the bad affect. On the other hand, the flow-based scheme, which is a distributed way to protect flows, could be ineffective or inefficient since the cooperation between each node in a network is more difficult than the cooperation between each scheduler in the hierarchical scheduler. From the point of view of protecting traffic flows to guarantee the requirement of QoS, the hierarchical scheduling scheme shows better performance than the distributed flow-based scheme.

It is also worth mentioning that the results have shown a trend of how a flow is affected by highly bursty traffic in a network using various scheduling schemes. In a real network, the actual values will be very likely to differ from the ones shown in these figures. What is important is the relative relation demonstrated by the results.

V. CONCLUSION

In this paper, we have proposed a topology-based hierarchical scheduling scheme for IPTV traffic management in Carrier Ethernet transport networks.

The hierarchical scheduler can be placed at the edge of broadband access network, where the topology is relatively static from IPTV dissemination point of view. Based on the assumption that the topology-based hierarchical scheduler is able to acquire the network topology, we have demonstrated a method that the hierarchical scheduler combines several DRR token schedulers to build a mapping structure of the connected network. The hierarchical scheduler manages traffic on behalf of other nodes in the network and is able to avoid severe performance degradation of normal traffic flows from maliciously behaving flows.

Simulation results have shown that the proposed scheduler can provide a better flow protection and isolation against potential attack from malicious traffic and as a result provide QoS guarantee, which is a significant requirement for IPTV services in Carrier Ethernet transport networks. The proposed scheme could also bring benefit to network operators in terms of deployment and cost-efficiency.

It is also important to mention that the hierarchical scheduling scheme presented in this paper is not limited to the topology used in the example. As a matter of fact, the scheduler can adapt to different network topologies. By network management or manual configuration, the scheduler can know where the potential congestion points are and how the network topology is. Different knowledge about the network leads to

different combination of the DRR token schedulers. The flexibility thus enables the scheduler to adapt to various network topologies, e.g. star, asymmetric tree and so forth. It is out of the scope of this paper to discuss how the combination of the DRR token schedulers is implemented. However, the research of such aspects will be carried out and continued in the future.

REFERENCES

- [1] Andy Reid, Peter Willis, Ian Hawkins, and Chris Bilton, "Carrier Ethernet", *IEEE Communication Magazine*, Sept. 2008.
- [2] M. Shreedhar and George Varghese, "Efficient fair queuing using deficit round-robin", *IEEE/ACM Transactions on Networking*, vol. 4, No. 3, Jun. 1996.
- [3] Hao Yu, Ying Yan, and Michael S. Berger, "IPTV traffic management in Carrier Ethernet transport networks", in *Proc. OPNETWORK 2008*, Aug. 2008.
- [4] Chin-Chi Wu, Hsien-Ming Wu, and Woei Lin, "Delivering relative differentiated services in future high-speed networks using hierarchical dynamic deficit round robin", *Multimedia Systems*, vol. 13, issue 3, pp. 205-221, 2007.
- [5] Doo-sung Back, Kihyun Pyun, Seung-Min Lee, Junhee Cho, and Namsu Kim, "A hierarchical deficit round-robin scheduling algorithm for a high level of fair service", in *Proc. 2007 International Symposium on Information Technology Convergence*, pp. 115-119, 2007.
- [6] Soranun Jiwassurat, George Kesidis, and David Miller, "Hierarchical shaped deficit round-robin scheduling", in *Proc. GLOBECOM '05. IEEE Global Telecommunications Conference*, vol. 2, pp. 689-693, 2005.
- [7] Chin-Chi Wu, Hsien-Ming Wu, and Woei Lin, "Delivering relative differentiated services in future high-speed networks using hierarchical dynamic deficit round robin", *Multimedia Systems*, vol. 13, issue. 3, pp. 205-221, 2007.
- [8] Mouad Ben Mamoun, Jean-Michel Fourneau, and Nihal Pekergin, "Analyzing weighted round robin policies with a stochastic comparison approach", *Computers & Operations Research* 35, pp. 2420-2431, Jan. 2007.
- [9] Abhay K. Parekh and RoberG. Gallagher, "A generalized processor sharing approach to flow control in integrated services networks: the single-node case", *IEEE/ACM Transactions on Networking*, vol. 1, issue 3, pp.344-357, 1993.
- [10] Jon C.R. Bennett and Hui Zhang, "WF²Q: worst-case fair weighted fair queueing", in *Proc. IEEE INFOCOM '96. Conference on Computer Communications*, vol. 1, pp 120-128, 1996.
- [11] S. Jamaloddin Golestani, "A self-clocked fair queueing scheme for broadband applications", in *Proc. INFOCOM '94 Conference on Computer Communications*, vol.2, pp 636-646, 1994.
- [12] P. Goyal, H.M. Vin, and H. Cheng, "Start-time fair queueing: a scheduling algorithm for integrated services packet switching networks", *IEEE/ACM Transactions on Networking*, vol. 5, issue 5, pp. 690-704, 1997.
- [13] T. Al-Khasib, H. Alnuweiri, H. Fattah, and V.C.M. Leung, "Fair and efficient frame-based scheduling algorithm for multimedia networks", in *Proc. the 10th IEEE Symposium on Computers and Communications*, 2005.
- [14] S. S. Kanhere, H. Sethu, and A. B. Parekh, "Fair and efficient packet scheduling using Elastic Round Robin", *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, issue. 3, pp. 324-336, 2002.
- [15] Working document: IPTV Multicast Frameworks, FG IPTV-DOC-0124, ITU-T Focus Group on IPTV, 2007
- [16] OPNET Modeler 14.0. <http://www.opnet.com>